AMENDMENTS TO THE CLAIMS

Please amend Claims 1 and 16 as follows:

1. (Currently Amended) A process for routing packets through a load balancing array of servers across a network in a computer environment, comprising the steps of:

requesting, by a scheduler, assignment of a virtual IP address to the scheduler, the scheduler is designated as active scheduler for [[a]] the load balancing array of servers;

wherein all incoming packets over a network from requesting clients destined for the load balancing array of servers are routed through the scheduler via the virtual IP address;

in response to receiving a request packet from a requesting client at the scheduler, routing and load balancing the request packet to a load balancing server, among a plurality of load balancing servers;

in response to receiving the request packet at the load balancing server, routing and load balancing the request packet to a back end Web server;

wherein the back end Web server's response packet to the request packet is sent to the load balancing server;

in response to receiving the response packet at the load balancing server, sending the response packet directly to the requesting client;

the sending step further comprising:

parsing, by the load balancing server, outgoing HTML page(s) ~~pages~~ in the response packet to determine selected content ~~to be~~ served by a content delivery network; and

modifying, by the load balancing server, URLs for the selected content in an HTML page in [[a]] the response packet in order to serve the selected content from the content delivery network in response to requests from requesting clients.

2. (Previously Presented) The process of Claim 1, wherein the scheduler is a load balancing server and routes and load balances client requests to itself.

3. (Previously Presented) The process of Claim 1, further comprising the steps of:

detecting failure of the scheduler; and

electing a load balancing server among a plurality of load balancing servers as a new scheduler.

4. (Previously Presented) The process of Claim 1, wherein the scheduler detects the failure of any load balancing servers among a plurality of load balancing servers in the load balancing array; and wherein the scheduler stops routing packets to any failed load balancing servers.

5. (Previously Presented) The process of Claim 1, wherein the load balancing server schedules sessions to back end Web servers based on a cookie or session ID.

6. (Previously Presented) The process of Claim 1, wherein the load balancing server uses cookie injection to map a client to a specific back end Web server.

7. (Previously Presented) The process of Claim 1, wherein the load balancing server decrypts a request packet in an SSL session before routing and load balancing the request packet to a back end Web server.

8. (Previously Presented) The process of Claim 7, wherein the load balancing server encrypts a response packet in an SSL session before sending the response packet directly to the requesting client.

9. (Previously Presented) The process of Claim 1, wherein the load balancing server establishes a connection with the requesting client and the requesting client keeps the connection alive with the load balancing server.

10. (Previously Presented) The process of Claim 9, wherein the load balancing server performs URL based scheduling of request packets.

11. (Previously Presented) The process of Claim 9, wherein the load balancing server performs hash scheduling of request packets.

12. (Previously Presented) The process of Claim 1, wherein the load balancing server maintains persistent connections in paths requiring persistent connections; and wherein the load balancing server uses hash group based persistence to maintain its persistence tables.

13. (Previously Presented) The process of Claim 1, wherein the load balancing server detects when a back end Web server fails; and wherein the load balancing server stops routing request packets to failed back end Web servers.

14. (Canceled)

15. (Previously Presented) The process of Claim 1, wherein HTML pages that have modified URLs are cached to improve performance.

16. (Currently Amended) An apparatus for routing packets through a load balancing array of servers across a network in a computer environment, comprising:

a scheduler, the scheduler requests assignment of a virtual IP address to the scheduler, the scheduler is designated as active scheduler for [[a]] the load balancing array of servers;

wherein all incoming packets over a network from requesting clients destined for the load balancing array of servers are routed through the scheduler via the virtual IP address;

wherein the scheduler routes and load balances a request packet from a requesting client to a load balancing server, among a plurality of load balancing servers;

wherein the load balancing server routes and load balances the request packet to a back end Web server;

wherein the back end Web server's response packet to the request packet is sent to the load balancing server;

wherein the load balancing server sends the response packet directly to the requesting client;

a module for parsing, by the load balancing server, outgoing HTML page(s) ~~pages~~ in the response packet to determine selected content ~~to be~~ served by a content delivery network; and

a module for modifying, by the load balancing server, URLs for the selected content in an HTML page in [[a]] the response packet in order to serve the selected content from the content delivery network in response to requests from requesting clients.

17. (Previously Presented) The apparatus of Claim 16, wherein the scheduler is a load balancing server and routes and load balances client requests to itself.

18. (Previously Presented) The apparatus of Claim 16, further comprising:

a module for detecting failure of the scheduler; and

a module for electing a load balancing server among a plurality of load balancing servers as a new scheduler.

19. (Previously Presented) The apparatus of Claim 16, wherein the scheduler detects the failure of any load balancing servers among a plurality of load balancing servers in the load balancing array; and wherein the scheduler stops routing packets to any failed load balancing servers.

20. (Previously Presented) The apparatus of Claim 16, wherein the load balancing server schedules sessions to back end Web servers based on a cookie or session ID.

21. (Previously Presented) The apparatus of Claim 16, wherein the load balancing server uses cookie injection to map a client to a specific back end Web server.

22. (Previously Presented) The apparatus of Claim 16, wherein the load balancing server decrypts the request packet when it is an SSL session before routing and load balancing the request packet to a back end Web server.

23. (Previously Presented) The apparatus of Claim 22, wherein the load balancing server encrypts the response packet when it is an SSL session before sending the response packet directly to the requesting client.

24. (Previously Presented) The apparatus of Claim 16, wherein the load balancing server establishes a connection with the requesting client and the requesting client keeps the connection alive with the load balancing server.

25. (Previously Presented) The apparatus of Claim 24, wherein the load balancing server performs URL based scheduling of request packets.

26. (Previously Presented) The apparatus of Claim 24, wherein the load balancing server performs hash scheduling of request packets.

27. (Previously Presented) The apparatus of Claim 16, wherein the load balancing server maintains persistent connections in paths requiring persistent connections; and

wherein the load balancing server uses hash group based persistence to maintain its persistence tables.

28. (Previously Presented) The apparatus of Claim 16, wherein the load balancing server detects when a back end Web server fails; and wherein the load balancing server stops routing request packets to failed back end Web servers.

29. (Canceled)

30. (Previously Presented) The apparatus of Claim 16, wherein HTML pages that have modified URLs are cached to improve performance.